

Refine Search

Search Results -

Terms	Documents
L4 AND ((unit ADJ test)OR (unit ADJ testing))	14

Database:

US Pre-Grant Publication Full-Text Database
 US Patents Full-Text Database
 US OCR Full-Text Database
 EPO Abstracts Database
 JPO Abstracts Database
 Derwent World Patents Index
 IBM Technical Disclosure Bulletins

Search:

L5

Refine Search

Recall Text

Clear

Interrupt

Search History

 DATE: Monday, December 13, 2004 [Printable Copy](#) [Create Case](#)
Set Name Query

side by side

Hit Count Set Name

result set

DB=USPT; PLUR=NO; OP=OR

<u>L5</u>	L4 AND ((unit ADJ test)OR (unit ADJ testing))	14	<u>L5</u>
<u>L4</u>	L3 AND (API OR Foci OR focus)	158	<u>L4</u>
<u>L3</u>	L2 Or 11	682	<u>L3</u>
<u>L2</u>	717/124.ccls.	269	<u>L2</u>
<u>L1</u>	Software ADJ testing	444	<u>L1</u>

END OF SEARCH HISTORY

Hit List

[Clear](#)[Generate Collection](#)[Print](#)[Fwd Refs](#)[Bkwd Refs](#)[Generate OACS](#)

Search Results - Record(s) 1 through 14 of 14 returned.

☐ 1. Document ID: US 6742177 B1

L5: Entry 1 of 14

File: USPT

May 25, 2004

US-PAT-NO: 6742177

DOCUMENT-IDENTIFIER: US 6742177 B1

TITLE: Method and system for secure debugging of a secure software module

DATE-ISSUED: May 25, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dorak, Jr.; John J.	Boca Raton	FL		
Cagle; Stephen B.	Coral Springs	FL		
Spagna; Richard L.	Boca Raton	FL		

US-CL-CURRENT: 717/124

ABSTRACT:

A method to debug application interface calls made to a tamper-resistant software module. The method comprises the steps of: loading a first application to be debugged using a debugging application, wherein the application makes one or more function calls to a tamper-resistant software module. A debugger application for symbolically debugging the first application is run and the debugger application with an initialization file that performs the steps of: loading a client application running a client socket service; loading a server application running a server socket service; loading the tamper-resistant software module with one or more function calls made thereto by the first application. The client application translates the one or more function calls made to the tamper-resistant software module during the running of the debugger application into socket calls and returns any status received from the socket calls to the first application. The server application translates the one or more function calls made to the tamper-resistant software module that are received via socket calls from the client application into function calls into the tamper-resistance software module and returning any status received back from the function calls into socket calls for receipt for the server application.

In an alternate embodiment, a computer readable medium and system is disclosed that corresponds to the method above.

16 Claims, 5 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 4

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw. D
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	---------

☐ 2. Document ID: US 6725399 B1

L5: Entry 2 of 14

File: USPT

Apr 20, 2004

US-PAT-NO: 6725399

DOCUMENT-IDENTIFIER: US 6725399 B1

TITLE: Requirements based software testing method

DATE-ISSUED: April 20, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman; John	Dallas	TX		

US-CL-CURRENT: 714/38; 717/124

ABSTRACT:

A method for testing computer software is described. The method is intended for operation on software which may or may not have been previously subjected to unit and integration tests, to determine if it will meet the specific requirements of an end user. The method is applicable to internally developed, contractor developed, or vendor supplied software. The method includes application in six key process areas. The six key process areas are development of a test plan, development of test cases to support the plan, development of an environment to simulate the technical environment in which the program will operate, test execution in which the tests are executed in a technical environment, compiling and analyzing the results and finally reporting the results in a form whereby the end user can determine both the feasibility of the software system for the specific requirements and any areas where additional testing or modifications are necessary.

9 Claims, 20 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 19

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw. D
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	---------

☐ 3. Document ID: US 6701514 B1

L5: Entry 3 of 14

File: USPT

Mar 2, 2004

US-PAT-NO: 6701514

DOCUMENT-IDENTIFIER: US 6701514 B1

TITLE: System, method, and article of manufacture for test maintenance in an automated scripting framework

DATE-ISSUED: March 2, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Haswell; John Jeffrey	Herndon	VA		
Young; Robert J.	Charlestown	MA		
Schramm; Kevin	Rose Valley	PA		

US-CL-CURRENT: 717/115; 707/102, 717/124*Scripting*

ABSTRACT:

A system, method and article of manufacture are provided for affording test maintenance in an automated scripting framework. First, a plurality of test scripts are developed. Then, the plurality of test scripts are stored in a centrally located database. A user is then allowed to edit a specific test script located on the centrally located database. Finally, the user edits to the specific test script are propagated to each of the plurality of test scripts.

18 Claims, 82 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 52

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KMHC	Draw D
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	--------

☐ 4. Document ID: US 6601018 B1

L5: Entry 4 of 14

File: USPT

Jul 29, 2003

US-PAT-NO: 6601018

DOCUMENT-IDENTIFIER: US 6601018 B1

TITLE: Automatic test framework system and method in software component testing

DATE-ISSUED: July 29, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Logan; Timothy Dennis	Walnut Creek	CA		

US-CL-CURRENT: 702/186; 714/38, 717/124

ABSTRACT:

Method and system aspects for test execution for software components within the context of an automatic test framework are described. In a method aspect, the method includes reading an executable file of a component, executing a test case code generator automatically on the executable file, and generating a skeleton test suite as a base for an individualized test case of the component. In addition, the method includes reading a JAVA class file, and utilizing an introspection facility.

12 Claims, 6 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 4

*No mention of
Focus-API
closest*

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	RWMC	Draw. De
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	----------

☐ 5. Document ID: US 6574578 B1

L5: Entry 5 of 14

File: USPT

Jun 3, 2003

US-PAT-NO: 6574578

DOCUMENT-IDENTIFIER: US 6574578 B1

TITLE: Server system for coordinating utilization of an integrated test environment for component testing

DATE-ISSUED: June 3, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Logan; Timothy Dennis	Walnut Creek	CA		

US-CL-CURRENT: 702/122; 709/203, 719/319

ABSTRACT:

Method and system aspects for utilizing a server to coordinate component testing in an integrated test environment network are described. In a method aspect, the method includes accepting a logon attempt by a client system and requesting a desired partition for the client system from a partition server. The method further includes reporting results from the client system on a database server, supplying report data to a Web server, and logging the client system off the server.

14 Claims, 6 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 4

Same as prior

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	RWMC	Draw. De
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	----------

☐ 6. Document ID: US 6510402 B1

L5: Entry 6 of 14

File: USPT

Jan 21, 2003

US-PAT-NO: 6510402

DOCUMENT-IDENTIFIER: US 6510402 B1

TITLE: Component testing with a client system in an integrated test environment network

DATE-ISSUED: January 21, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY

NOT unit testing

Logan; Timothy Dennis Walnut Creek CA
Ramaswamy; Pudur S. Cupertino CA

US-CL-CURRENT: 702/186; 709/203, 714/38, 719/319

ABSTRACT:

Method and system aspects for performing component testing with a client system in an integrated test environment network are described. In a method aspect, the method includes retrieving one or more test suites and associated platform partitions from at least one server system to initiate component testing. The method further includes performing test cases for the test suites, and providing results of the test cases to the at least one server system.

16 Claims, 6 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 4

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw. D.
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	----------

☐ 7. Document ID: US 6505342 B1

L5: Entry 7 of 14

File: USPT

Jan 7, 2003

US-PAT-NO: 6505342

DOCUMENT-IDENTIFIER: US 6505342 B1

TITLE: System and method for functional testing of distributed, component-based software

DATE-ISSUED: January 7, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Hartmann; Jean S.	Robbinsville	NJ		
Imoberdorf; Claudio	Princeton	NJ		

prior

US-CL-CURRENT: 717/104; 717/107, 717/125

ABSTRACT:

A system and method for functional testing of distributed, component-based software that includes modeling dynamic behavior of a plurality of components using a plurality of Unified Modeling Language-based state machine representations; creating a plurality of normalized Unified Modeling Language-based state machine representations by converting the Unified Modeling Language based state machine representations into a normalized representation; creating a global behavioral model by combining the normalized Unified Modeling Language based state machine representations; mapping the global behavioral model to a Test Specification Language (TSL) test design; processing the Test Specification Language test design in a Test Design Environment to produce a plurality of Interface Test Language (ITL) test cases; mapping the ITL test cases to standard object-oriented code using an ITL compiler; generating a plurality of sink objects defined in an IDL file

using a sink generator; generating an executable test driver containing a plurality of executable test cases using code from a test harness library and the above generated code using a standard object-oriented compiler; and executing the test cases with a test control center.

22 Claims, 18 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 14

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KMIC	Draw D
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	--------

☐ 8. Document ID: US 6502102 B1

L5: Entry 8 of 14

File: USPT

Dec 31, 2002

US-PAT-NO: 6502102

DOCUMENT-IDENTIFIER: US 6502102 B1

**** See image for Certificate of Correction ****

TITLE: System, method and article of manufacture for a table-driven automated scripting architecture

DATE-ISSUED: December 31, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Haswell; John Jeffrey	Herndon	VA		
Young; Robert J.	Charlestown	MA		
Schramm; Kevin	Rose Valley	PA		

US-CL-CURRENT: 707/102

ABSTRACT:

A system, method and article of manufacture are provided for affording a table-driven automated scripting architecture. First, test script information is divided into a plurality of components of one or more words having a commonly understood meaning. Then the components are stored into a database. Later, the components are parsed into one or more words (each having a commonly understood meaning). The database is queried for the words to retrieve a set of computer instructions that cause a computer to perform functions related to the commonly understood meaning of the words and then to perform those functions.

20 Claims, 82 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 51

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KMIC	Draw D
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	--------

☐ 9. Document ID: US 6405364 B1

L5: Entry 9 of 14

File: USPT

Jun 11, 2002

US-PAT-NO: 6405364

DOCUMENT-IDENTIFIER: US 6405364 B1

TITLE: Building techniques in a development architecture framework

DATE-ISSUED: June 11, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah, Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 717/101; 717/102, 717/120, 717/124

ABSTRACT:

A system is provided for building systems in a development architecture framework. The present invention is directed to both a system to be built and an implementation strategy to fulfill system requirements. Software components of the system are encapsulated with wrappers. The wrappers are adapted to be changed upon other software components of the system being changed while the encapsulated software components of the system remain unchanged. In one embodiment of the present invention, specifying the requirements of the system to be built and the implementation strategy to fulfill the requirements may be carried out using tools such as data modeling tools, process modeling tools, event modeling tools, performance modeling tools, object modeling tools, component modeling tools, reuse support tools, prototyping tools, application logic design tools, database design tools, presentation design tools, communication design, and usability test tools. In another embodiment of the present invention, improving the performance and maintenance of the system may be carried out using tools such as interactive navigation tools, graphical representation tools, extraction tools, repository tools, restructuring tools, and data name rationalization tools.

*Integrated in
Dev Env
main +
wrappers
for
sys
req*

12 Claims, 14 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 14

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw D
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	--------

☐ 10. Document ID: US 6028999 A

L5: Entry 10 of 14

File: USPT

Feb 22, 2000

US-PAT-NO: 6028999

DOCUMENT-IDENTIFIER: US 6028999 A

TITLE: System and method for non-sequential program statement execution with incomplete runtime information

DATE-ISSUED: February 22, 2000

INVENTOR-INFORMATION:

*Disturbance
Testing*

NAME	CITY	STATE	ZIP CODE	COUNTRY
Pazel; Donald Philip	Montrose	NY		

US-CL-CURRENT: 717/134; 712/227, 714/25, 714/37, 714/38

ABSTRACT:

A system and method for non-sequential program execution in an incompletely assembled runtime environment. In a program debugging context, a primary advantage is that neither a fully linked executable nor a complete code base is required to execute the program. As well, it is not necessary to begin execution at the traditional program entry point. Six key methods are described. Virtual execution initialization logic involves establishing the runtime environment within the context of the first statement of execution. Statement execution logic concerns execution of a program statement and a technique by which the runtime copes with incomplete information. Reset execution point logic allows the user to randomly reposition the current execution point within a program without violating the runtime environment. Finally, methods for adding and deleting callers into/from the runtime stack are outlined. Extensions for multi-process and thread support are outlined as well as the incorporation of standard and application libraries into the execution model.

40 Claims, 13 Drawing figures
Exemplary Claim Number: 23
Number of Drawing Sheets: 13

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWOC	Draw D
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	--------

☐ 11. Document ID: US 5712972 A

L5: Entry 11 of 14

File: USPT

Jan 27, 1998

US-PAT-NO: 5712972

DOCUMENT-IDENTIFIER: US 5712972 A

**** See image for Certificate of Correction ****

TITLE: Identification of faults in data paths and functional units of a central processing unit by a systematic execution of test instructions

DATE-ISSUED: January 27, 1998

INVENTOR- INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Kakkar; Sunil	San Jose	CA		

US-CL-CURRENT: 714/26

ABSTRACT:

A systematically structured diagnostic for detecting, isolating, analyzing and reporting problems or faults in a central processing unit. The diagnostic causes the central processing unit to execute instructions, the instructions being selected such that the central processing unit, in executing the instructions, must

Fault isolation

use every data path and functional unit therein. Errors caused by particular instructions are correlated with the functional units or datapaths used by those instructions, to produce a list of possibly faulty datapaths and functional units. Test procedures, specifically designed to the possibly faulty datapaths and functional units, are then applied to the central processing unit to isolate which of the possibly faulty data paths or functional units are in fact faulty, which are then reported. The instructions, test procedures and other information used by the diagnostic are stored in databases, so that the diagnostic has a modular and data-driven structure which permits evolution of the diagnostic over time as the central processing unit layout changes and the diagnostic is upgraded to provide higher levels of fault-detection functionality.

21 Claims, 1 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 1

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw. D
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	---------

☐ 12. Document ID: US 5708774 A

L5: Entry 12 of 14

File: USPT

Jan 13, 1998

US-PAT-NO: 5708774
DOCUMENT-IDENTIFIER: US 5708774 A

TITLE: Automated testing of software application interfaces, object methods and commands

DATE-ISSUED: January 13, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Boden; Edward Barnes	Vestal	NY		

US-CL-CURRENT: 714/38; 706/13, 706/922

ABSTRACT:

Automated testing of software application interfaces, object methods and commands, including testing software using order-based genetic algorithms to search for and detect symptoms of software errors by generating test sequences which converge on points in invocation space more likely to cause error symptoms.

12 Claims, 6 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 3

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequence	Attachments	Claims	KWIC	Draw. D
------	-------	----------	-------	--------	----------------	------	-----------	----------	-------------	--------	------	---------

☐ 13. Document ID: US 5651111 A

L5: Entry 13 of 14

File: USPT

Jul 22, 1997

US-PAT-NO: 5651111

DOCUMENT-IDENTIFIER: US 5651111 A

TITLE: Method and apparatus for producing a software test system using complementary code to resolve external dependencies

DATE-ISSUED: July 22, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
McKeeman; William M.	Hollis	NH		
Reinig; August G.	Hudson	NH		

*Prior
not behavior
model*

US-CL-CURRENT: 714/38; 717/126

ABSTRACT:

A software unit development and test methodology in which a software application or project is dividing into conceptual units. Each unit is first developed and debugged in an isolated testing environment which simulates the actual testing environment through test conditions. Following unit testing, other tested units are incrementally combined and tested in a similar isolated manner. Automatic generation of a testing environment and development system driving debugging and testing software, for measuring testing completeness, and for verifying correctness of future development and maintenance efforts are provided.

29 Claims, 4 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 4

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw D
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	--------

☐ 14. Document ID: US 5490249 A

L5: Entry 14 of 14

File: USPT

Feb 6, 1996

US-PAT-NO: 5490249

DOCUMENT-IDENTIFIER: US 5490249 A

TITLE: Automated testing system

DATE-ISSUED: February 6, 1996

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Miller; Christopher M.	Saratoga	CA		

US-CL-CURRENT: 714/38; 714/26, 714/39

ABSTRACT:

A method and apparatus for testing software programs systematically explores valid call sequences using a collection of data with a software program comprising a collection of subroutine components. Instead of writing unit tests, the subject invention provides tools to develop precondition rules which systematically remove from the data collection, data which is syntactically incorrect for proper operation with the software program and then data which is semantically incorrect for the software program. The resulting reduced data collection is applied to the collection of components to generate output values which are then checked against postcondition rules to verify that the software program operated correctly. The test method and apparatus are implemented in object-oriented technology to allow test engineers to quickly and easily develop invariants, precondition rules and postcondition rules which are used to test the software program.

7 Claims, 7 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 7

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw. Des.
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	------------

Clear	Generate Collection	Print	Fwd Refs	Bkwd Refs	Generate OACS
-------	---------------------	-------	----------	-----------	---------------

Terms	Documents
L4 AND ((unit ADJ test)OR (unit ADJ testing))	14

Display Format: REV Change Format

[Previous Page](#) [Next Page](#) [Go to Doc#](#)